



NRL/MR/5320-97-7929

## Porting a Ptolemy-Based Simulation Between Sites

A. R. CAUL-TOSKIN

*Advanced Radar Systems Branch  
Radar Division*

April 14, 1997

DTIC QUALITY INSPECTED

Approved for public release; distribution unlimited.

19970410 044

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE  April 14, 1997		3. REPORT TYPE AND DATES COVERED
4. TITLE AND SUBTITLE  Porting a Ptolemy-Based Simulation Between Sites			5. FUNDING NUMBERS	
6. AUTHOR(S)  Athena R. Caul-Toskin				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Naval Research Laboratory Washington, DC 20375-5320			8. PERFORMING ORGANIZATION REPORT NUMBER  NRL/MR/5320-97-7929	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Defense Advanced Research Projects Agency 3701 North Fairfax Drive Arlington, VA			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE  A	
13. ABSTRACT (Maximum 200 words)  The intent of this report is to document the successful testing of some RASSP products produced on the Navy monitored DARPA/Tri-Service Lockheed Sanders RASSP contract and to document the process of porting a Ptolemy-based simulation between two sites. It is beneficial to have this capability to build on and reuse previous simulations and designs done by supporting institutions and to learn from others' work with the goal of decreasing development times and not repeating work already done. The process used to port an infrared search and track (IRST) simulation between two sites is explained as well as the design environment in which the simulation was built, Ptolemy. The difficulties encountered in porting this particular simulation between these two sites are explained. Some implications regarding the porting process and its future use are drawn from the knowledge gained from this endeavor. In conclusion, the simulation was successfully ported between the two sites.				
14. SUBJECT TERMS  Ptolemy Porting  Simulation Port			15. NUMBER OF PAGES  9	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT  UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE  UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT  UNCLASSIFIED	20. LIMITATION OF ABSTRACT  UL	

## TABLE OF CONTENTS

1. INTRODUCTION .....	1
2. DESIGN ENVIRONMENT: PTOLEMY .....	1
3. IRST SIMULATION.....	2
4. PORTING A PTOLEMY SIMULATION .....	2
4.1 Procedure.....	2
4.2 Difficulties Encountered.....	4
4.3 Porting Implications .....	4
5. SUMMARY .....	5
6. ACKNOWLEDGEMENTS .....	6
7. REFERENCES .....	6

# **PORTING A PTOLEMY-BASED SIMULATION BETWEEN SITES**

## **1. INTRODUCTION**

Presently, designers must utilize many resources to meet the demand for shorter design and development cycles. These include in particular: the utilization of simulation environments, the reuse of previous simulations and the reuse of design hardware and software. With the increasing importance of reusability and portability of both software and hardware, the ability to reuse portions of simulations and designs is essential. The ability to share previously completed simulations and designs with supporting institutions shortens secondary development cycles since a designer does not have to start from scratch. This report details how an infrared search and track (IRST) simulation created by Lockheed Sanders, Inc. (Sanders) was ported to a platform at the Naval Research Laboratory (NRL) as part of the DARPA/Tri-Service Rapid Prototyping of Application Specific Signal Processors (RASSP) contract monitoring process. The procedure used and the difficulties encountered in porting and verifying the simulation are discussed.

## **2. DESIGN ENVIRONMENT: PTOLEMY**

The IRST simulation was developed by Sanders under DARPA's RASSP program. The model was designed using the version 0.6 Ptolemy simulation and design software developed at the University of California at Berkeley. Ptolemy provides multiple design environments suitable for modeling signal processing systems, communication networks, circuit designs and software algorithms among others. Although the IRST simulation uses only one of these environments, Ptolemy allows the user to combine multiple environments to create a single heterogeneous design. Ptolemy is readily usable since many design components have already been created, but also allows users to add extensions useful in their particular applications. The IRST simulation is largely composed of Sanders user-created components.

Ptolemy also supports multiple computational models within its design environments such as data-dependent flow of control, discrete-event processing, sequential processes, etc. An in depth explanation of the numerous capabilities of Ptolemy is too complicated to present here. A thorough explanation can be found in the Ptolemy user's manual [1].

The Ptolemy terminology necessary to explain the IRST simulation is listed below (this is a minimal subset of all the important terminology related to Ptolemy):

- **Block.** The software modules which are combined to describe a system. A block can be either a Star or a Galaxy.
- **Star.** An elementary block implemented by user-provided code.
- **Galaxy.** A hierarchical block which contains Stars and possibly other Galaxies.
- **Universe.** A complete system.
- **Domain.** A model of computation e.g. dataflow.
- **Palette.** A graphical representation of a library of icons which represent Stars and Galaxies.

A graphical representation of a generic universe is shown in Figure # 1. A hierarchical system is represented as a Universe composed of Stars and Galaxies and shows the flow of information between these blocks. The hierarchical nature of a Galaxy is shown by displaying within the highlighted box the contents of a top level Galaxy. Each level of the design is viewed separately and the user can traverse the levels of the hierarchy to view the contents of each component Galaxy. Each XXXStar and Galaxy depicted is unique and the words Star and Galaxy in each are replaced with the function or name of the Star or Galaxy. The domain of a particular Star is represented by 3 letters preceding the Star name i.e. an XXXAdd star is from the XXX domain and performs an addition function.

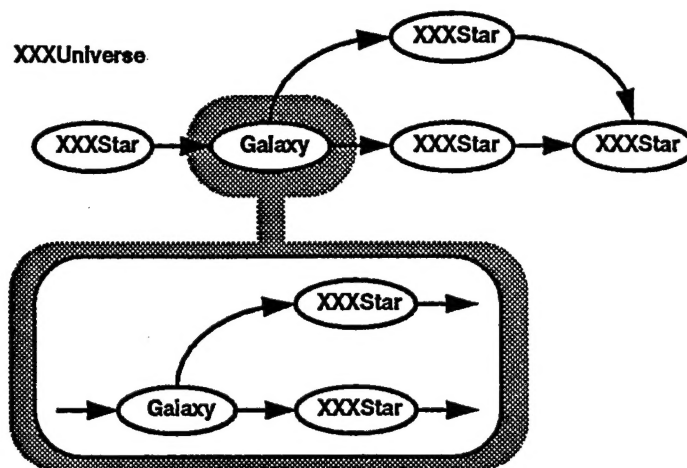


Figure # 1 - A graphical representation of a generic universe

### 3. IRST SIMULATION

The IRST simulation utilizes the synchronous dataflow (SDF) domain as its signal processing design environment. The information necessary to re-create the IRST simulation consists of multiple user-created stars written in C++, including the icons that represent these stars, the graphical representation of the IRST universe and the data for the simulation.

### 4. PORTING A PTOLEMY SIMULATION

#### 4.1 Procedure

The first step is to ensure that the same version of Ptolemy (and its source code) is installed at both sites since designs may not be compatible with differing versions. Users who wish to add more than a few stars should statically link their additions to the Ptolemy kernel although dynamic linking is possible but is not efficient. Users adding substantial extensions, such as adding a new domain, must statically link these extensions as dynamic linking is not possible. Static linking consists of adding the code for the extensions to the Ptolemy code and compiling the additions into the Ptolemy executable. Dynamic linking is possible when only a few stars are being added and consists of compiling the code for the new stars separately and linking these to the Ptolemy executable. Dynamic linking is not preferred for adding a large number of stars since dynamic linking occurs at execution time and the user must wait for the

linking to occur each time one of these stars is executed. The Ptolemy programmer's manual [2] provides a detailed explanation of static and dynamic linking and how each is accomplished. For the case of the IRST simulation, the new stars were statically linked which required compiling Ptolemy from its source code.

To compile Ptolemy Version 0.6 from source, the following supporting tools (of which all except for X Windows are available from the Ptolemy Web site [3] ) are necessary:

- gcc compiler - version 2.7.2
- libg++ library - version 2.7.1
- GNU make - version 3.74
- octtools
- itcl - version 2.0 which includes Tcl version 7.4p3 and Tk version 4.0p3
- xv sources
- X Window system version 11 release 5 (X11R5)

Particular attention should be paid to installing the correct version of the tools as Ptolemy is very sensitive to these. When the Ptolemy source code is compiled for the first time, the compilation takes several hours and uses a large (approximately 300 MBytes) amount of disk space. Once all of the Ptolemy source code has been compiled initially, any future statically linked extensions require only an incremental compilation which compiles relatively quickly.

The path name where Ptolemy is installed is referred to as \$PTOLEMY. To compile the source code, invoke the command 'make install' at the Ptolemy installation directory, \$PTOLEMY. After the basic compilation is completed, user-created extensions can be installed.

New stars can be added to an existing domain directory, to a new directory, or within a new user-created domain. In this case, the new stars and their icons are added to the matrix operations section of the existing SDF domain directory. Therefore, the icon files are placed in the directory \$PTOLEMY/src/domains/sdf/matrix/icons. The source code for the stars (\*.cc and \*.h files) is placed in the directory \$PTOLEMY/src/domains/sdf/matrix/stars. In the star source code directory, the makefile template must be copied over the makefile and a 'make depend' command issued to place the new stars in the makefile with their dependencies. After these steps, an incremental compile can be made by again invoking the command 'make install' in the directory \$PTOLEMY. After this compilation is completed, the new stars are statically linked to the Ptolemy kernel.

The IRST simulation itself must be set up in a user or a project working directory. The working directory would include palette files (\*.pal) containing the graphical representation of the system and the input data files (\*.dat) for the simulation. The path names within each palette must be changed to the correct site-specific paths. To change path names within the palettes, the Ptolemy "masters" program must be run on each palette. There may also be incorrect path names within the parameters of each new star that will not be evident and cannot be modified until the simulation is displayed within Ptolemy. The ease of portability of the IRST simulation could be improved if the use of hard-coded or site-specific path names could be avoided. In general, designs are more portable when generic path names are used. This can be accomplished by substituting an environment variable for the hard-coded project working

directory. It is much easier for a user to set up an environment variable than to find and change all of the incorrect hard-coded or site-specific path names.

The IRST simulation contains five palette files, one input data file and four weighting data files. Five path names within the palette files were changed using the "masters" program. A total of 16 path names were changed within the star parameters when the top level palette was displayed within Ptolemy. The 16 path names changed included eleven for output files, one for the input file and four for the weight files. These path names could be replaced by a single environment variable defining the base project working directory and the file name and directory beneath that base directory, e.g., the new path name for a generic output file would be \$IRST\_PROJECT\_DIR/output\_dir/filename.dat. In the future, users who port this simulation would just need to create the base working directory and its subdirectories and define the environment variable IRST\_PROJECT\_DIR.

In conclusion, all of these steps were successfully completed and the simulation was successfully run at the NRL site.

## **4.2 Difficulties Encountered**

The majority of the problems encountered in this project were related to the strict version dependencies of the supporting tools and the compilation of the source code. Another complicating factor at the NRL site was the existence of multiple versions of some of the supporting software and libraries. At a site where the correct Ptolemy-specific versions of the libraries or X Windows are not normally used, links must be set up so Ptolemy will look for its version while the users at the site continue to work with the normal site version. A related problem is ensuring that the ordering of the libraries and "#include" files enables Ptolemy to find its version of tools first before finding a possibly incorrect site version. There are also a number of environment variables which also deal with setting up the path names to certain files or tools that must be manipulated. The modification of some make macros in the configuration files was also necessary. The length of time necessary to compile the source code was also a hindrance. An error was sometimes not evident for a matter of hours, at which point the compilation would have to be begun again. A complete compilation takes approximately eight hours to complete on a Sun4/280.

## **4.3 Porting Implications**

After completing the procedures outlined above, all users working with the same site version of Ptolemy have access to and use of the new stars, galaxies and universes. Therefore, additional users can build on these additions by creating new galaxies and universes. Also users can add additional extensions more readily since the basic source code compilation has been completed and only incremental compilations are necessary.

If it becomes necessary to port this simulation to another site or platform, the new site will have to install the Ptolemy source code (not the prebuilt binaries that are available) and the same procedures will have to be repeated. The only savings that could be gained would be if the new site had already completed the initial source code compilation and the statically linked extensions could be incrementally compiled into the site version. It is doubtful that compiled code could be exchanged since the extensions are statically linked into the kernel and the same compiler must be used on the source code as on the extensions. Exchanging dynamically



linked stars would allow exchange of compiled code but would create a great run time penalty if a large number of stars are to be linked. A caveat of exchanging compiled code for dynamic linking is that the same compiler version must be used to compile Ptolemy as to compile the stars.

The use of makefiles within Ptolemy automates the majority of the installation of extensions. It is probable that the installation of extensions could be further automated through a script or by some other means although this is not really necessary when there are only a few steps to the installation. Some issues which complicate the generalization of installations include the varying complexity of extensions, the variety of installation directories and the use of hard-coded or site-specific path names.

The following resources may provide information on new developments, e.g. scripts, patches and bug fixes, as well as providing a forum for discussion of Ptolemy questions and issues:

- (a) the Ptolemy newsgroup (comp.soft-sys.ptolemy)
- (b) the ptolemy-hackers electronic mailing list<sup>1</sup> (at ptolemy-hackers@ptolemy.eecs.berkeley.edu) where users can post and read about Ptolemy questions and issues and
- (c) the ptolemy-interest electronic mailing list<sup>2</sup> (at ptolemy-interest@ptolemy.eecs.berkeley.edu) announces new releases, bug fixes and other information but is not a discussion group.

## 5. SUMMARY

The intent of this report is to document the successful testing of some RASSP products produced on the Navy monitored DARPA/Tri-Service Lockheed Sanders RASSP contract and to document the process of porting a Ptolemy-based simulation between two sites. It is beneficial to have this capability to build on and reuse previous simulations and designs done by supporting institutions and to learn from others' work with the goal of decreasing development times and not repeating work already done. The process used to port an infrared search and track (IRST) simulation between two sites is explained as well as the design environment in which the simulation was built, Ptolemy. The difficulties encountered in porting this particular simulation between these two sites are explained. Some implications regarding the porting process and its future use are drawn from the knowledge gained from this endeavor. In conclusion, the simulation was successfully ported between the two sites.

---

<sup>1</sup> To subscribe to the ptolemy-hackers mailing list, send electronic mail to ptolemy-hackers-request@ptolemy.eecs.berkeley.edu with the word "subscribe" in the body of the letter.

<sup>2</sup> To subscribe to the ptolemy-interest mailing list, send electronic mail to ptolemy-interest-request@ptolemy.eecs.berkeley.edu with the word "subscribe" in the body of the letter.



## **6. ACKNOWLEDGEMENTS**

The author would like to recognize Eric K. Pauer of Lockheed Sanders, Inc., who provided the IRST simulation and a set of porting instructions to NRL. This list of instructions provided the framework for the Procedures listed herein.

## **7. REFERENCES**

1. Ptolemy Working Group: "The Almagest Volume 1: Ptolemy 0.6 User's Manual", University of California at Berkeley, Department of Electrical Engineering and Computer Sciences.
2. Ptolemy Working Group: "The Almagest Volume 3: Ptolemy 0.6 Programmer's Manual", University of California at Berkeley, Department of Electrical Engineering and Computer Sciences.
3. Ptolemy World Wide Web Site: <http://ptolemy.eecs.berkeley.edu/>